(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平11－110504

(43)公開日　平成11年(1999)４月23日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｋ　17/00		Ｇ０６Ｋ　17/00	Ｔ
			Ｄ
			Ｌ
Ｂ４２Ｄ　15/10	５２１	Ｂ４２Ｄ　15/10	５２１
Ｇ０９Ｃ　1/00	６２０	Ｇ０９Ｃ　1/00	６２０Ｂ

審査請求　未請求　請求項の数６　ＯＬ　（全　８　頁）　最終頁に続く

(21)出願番号　特願平9－267324

(22)出願日　平成９年(1997)９月30日

(71)出願人　000003193
凸版印刷株式会社
東京都台東区台東１丁目５番１号

(71)出願人　000004226
日本電信電話株式会社
東京都新宿区西新宿三丁目19番２号

(72)発明者　山岡　憲一
東京都台東区台東１丁目５番１号　凸版印刷株式会社内

(72)発明者　平野　誠治
東京都台東区台東１丁目５番１号　凸版印刷株式会社内

(74)代理人　弁理士　川▲崎▼　研二

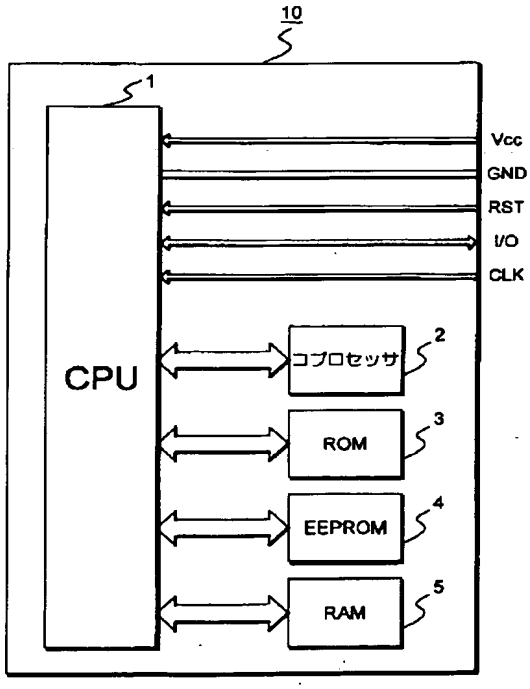
最終頁に続く

(54)【発明の名称】　ＩＣカード

(57)【要約】

【課題】　電子現金等のシステムで用いられた場合でも、各種データについての処理時間が短く、セキュリティ性の高いＩＣカードを提供する。

【解決手段】　ＲＡＭ５またはＥＥＰＲＯＭ４には、金融取引等についてのデータが複数のレコードに分割して格納してある。ＣＰＵ１は、その複数のレコードを所定のレコード数毎に結合してデータのかたまりをつくる。そして、ＣＰＵ１は、その結合したデータのかたまり毎に圧縮処理および暗号化処理を行う。そして、ＣＰＵ１は、その圧縮処理および暗号化処理した結果をＲＡＭ５またはＥＥＰＲＯＭ４に格納するとともに、署名データとして端末装置側に送出する。ここで、コプロセッサ２は、暗号化処理におけるべき乗余剰演算を行う。



【特許請求の範囲】
【請求項１】　外部装置と接続され、書き換え可能なメモリ手段と、外部からの命令に基づいてデータ処理を行う演算手段を有するＩＣカードにおいて、
前記メモリに格納されているデータであって複数のレコードに分割してあるデータを、所定数のレコード毎に１個のブロックに結合する結合手段と、
前記結合手段が結合したブロック毎に連続して当該データをデータ圧縮をする圧縮手段と、
前記圧縮手段によって圧縮したデータを前記メモリに格納する格納手段と、
前記格納手段によって格納したデータを所定の暗号方式で暗号化することで署名データを生成する署名データ生成手段とを具備することを特徴とするＩＣカード。
【請求項２】　請求項１記載のＩＣカードにおいて、
前記署名データを不揮発的に保持する署名データ保持手段を有することを特徴とするＩＣカード。
【請求項３】　請求項１または２記載のＩＣカードにおいて、
前記署名データ生成手段は、べき乗余剰演算が可能なコプロセッサを用いて暗号化の処理をすることを特徴とするＩＣカード。
【請求項４】　請求項１、２または３記載のＩＣカードにおいて、
前記圧縮手段は、一方向性ハッシュ関数を用いてデータ圧縮することを特徴とするＩＣカード。
【請求項５】　請求項１、２、３または４記載のＩＣカードにおいて、
前記署名データ生成手段における暗号方式は、公開型暗号方式であることを特徴とするＩＣカード。
【請求項６】　請求項１、２、３、４または５記載のＩＣカードにおいて、
前記メモリに格納されているデータであって複数のレコードに分割してあるデータは、取引の履歴の一部を示す取引データを含むことを特徴とするＩＣカード。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、電子現金等を取り扱うシステムで用いられるＩＣカードに関する。特に本発明は、取引の履歴を示すデータであるＬＯＧデータを用いてＩＣカード内で署名データを生成して、取引データの認証をするＩＣカードに関する。
【０００２】
【従来の技術】ＩＣカードは、例えば、セキュリティ性を確保しなければならない金融取引システム（電子現金システム）に用いられる。この種のＩＣカードでは、商取引の状態（日時、場所、金額など）を示す取引データを取り扱い、また、セキュリティ性を高めるため暗号化した署名データを用いてその取引についての認証を行うようにしている。
【０００３】ところで、従来においては、署名データは、ＩＣカード内に格納されているデータを端末装置側で読み出し、これを暗号化して再度ＩＣカードに格納していた。ここで、端末装置側とは、ＩＣカードの通信相手となる装置であって、金融取引システムなどにおける上位システムをいう。また、取引データについては、１レコードについて暗号化してＴＡＣ（Transaction Authentication Code）を生成し、これをＩＣカードに格納していた。この場合、取引データが複数のデータによって構成されるときは、レコード毎にＴＡＣを生成し、ＩＣカードに格納するか、端末装置側で取引データを圧縮し、暗号化した後にＩＣカードに格納していた。
【０００４】なお、ＴＡＣとは、ＤＥＳ(Data Encryption Standard)、ＦＥＡＬ(Fast DataEncryption Algorithm)利用の認証子生成検査法による暗号をいう。ここで、ＤＥＳとは、アルゴリズム公開型の共通鍵方式の暗号方式をいう。共通鍵方式とは、復号鍵が暗号化鍵に等しいか暗号化鍵から簡単に導ける暗号方式をいい、対称暗号系ともいう。共通鍵方式は、比較的小規模のプログラムやハードウエアを用いても高速な処理が可能で、実用性の高いものを構成しやすい。
【０００５】
【発明が解決しようとする課題】ところで、従来のＩＣカードにおいては、レコード単位にＭＡＣを生成するので、複数のレコードからなる取引データ（日時、場所、金額等）についてはＭＡＣ生成を複数回行う必要があった。
【０００６】また、端末装置側でＩＣカードを使用するときは、セキュリティ性を高めるためにＭＡＣデータを８バイトの長いデータにする。そして、複数のレコードをＩＣカードに格納する場合は、各レコード毎に長いＭＡＣデータが付加されているので、ＩＣカードが内蔵するメモリの使用量を増大させてしまう。また、この場合は、そのメモリの内容においてＭＡＣデータ以外の取引データの割合が少なくなり、ＩＣカードのメモリの有効利用上も問題があった。
【０００７】本発明は、このような背景の下になされたものであり、認証用データ等を少ない記憶容量で効率的に記憶することができるとともに、セキュリティ性の高いＩＣカードを提供することを目的とする。
【０００８】
【課題を解決するための手段】上述した課題を解決するために、請求項１記載の発明は、情報を書き換え可能に格納するメモリと、前記メモリに格納されているプログラムに基づいて当該ＩＣカードの動作を制御する中央処理装置とを具備するＩＣカードにおいて、前記メモリに格納されているデータであって複数のレコードに分割してあるデータを、所定数のレコード毎に１個のブロックに結合する結合手段と、前記結合手段が結合したブロック毎に連続して当該データをデータ圧縮をする圧縮手段

と、前記圧縮手段によって圧縮したデータを前記メモリに格納する格納手段と、前記格納手段によって格納したデータを所定の暗号方式で暗号化することで署名データを生成する署名データ生成手段とを具備することを特徴とする。

【０００９】また、請求項２記載の発明は、請求項１記載のＩＣカードにおいて、前記署名データを不揮発的に保持する署名データ保持手段を有することを特徴とする。

【００１０】また、請求項３記載の発明は、請求項１または２記載のＩＣカードにおいて、前記署名データ生成手段は、べき乗余剰演算が可能なコプロセッサを用いて暗号化の処理をすることを特徴とする。

【００１１】また、請求項４記載の発明は、請求項１、２または３記載のＩＣカードにおいて、前記圧縮手段は、一方向性ハッシュ関数を用いてデータ圧縮することを特徴とする。

【００１２】また、請求項５記載の発明は、請求項１、２、３または４記載のＩＣカードにおいて、前記署名データ生成手段における暗号方式は、公開型暗号方式であることを特徴とする。

【００１３】また、請求項６記載の発明は、請求項１、２、３、４または５記載のＩＣカードにおいて、前記メモリに格納されているデータであって、複数のレコードに分割してあるデータは、取引の履歴の一部を示す取引データを含むことを特徴とする。

【００１４】

【発明の実施の形態】以下、図面を参照して、この発明の実施形態について説明する。

Ａ：実施形態の構成

図１は、本発明の一実施形態であるＩＣカードの構成を示すブロック図である。図において、１は装置各部を制御するＣＰＵであり、２は暗号化処理におけるべき乗余剰演算をするコプロセッサである。３は、ＣＰＵ１の動作を規定するプログラムなどを記憶するＲＯＭであり、ＥＥＰＲＯＭ４はデータを書き換え可能に記憶する不揮発性メモリである。なお、ＣＰＵ１の動作を規定するプログラムはＥＥＰＲＯＭ４に記憶させてもよい。ＲＡＭ５は、データを一時的に格納する揮発性メモリである。

【００１５】ＣＰＵ１は、ＩＣカード１０が端末装置（図示せず）に挿入された状態において、電源Ｖｃｃ、クロックＣＬＫ、リセット信号ＲＳＴおよびグランド電位ＧＮＤが与えられるようになっている。また、ＣＰＵ１は、端末装置とＩＣカード１０との間の双方におけるデータの授受をＩ／Ｏ端子を介して行う。

【００１６】ＲＡＭ５またはＥＥＰＲＯＭ４の一部は、ＣＰＵ１が各種の情報処理をする際の作業バッファともなる。さらにＲＡＭ５またはＥＥＰＲＯＭ４の一部は、図６に示すようなファイル構造のレコードが格納されてる。そのレコードは、金融取引または商取引などにおける金額、取引時間、取引場所などの履歴を示すデータなどからなる。

【００１７】Ｂ：実施形態の動作

次に、上記構成からなるＩＣカード１０の動作を説明する。図２は、ＩＣカード１０におけるコマンド処理を示すフローチャートである。まず、ＣＰＵ１は、端末装置側から受けた命令がハッシュコマンドであるか否かを判断する（Ｓ１）。ここで、ハッシュコマンドである場合は、図３に示すレコード結合処理及び図４に示す圧縮処理をする（Ｓ２）。すなわち、ハッシュコマンドとは、ＩＣカード１０の内部において、複数のレコードを結合して、その結合したデータをハッシュ関数で圧縮処理することを要求するコマンドである。

【００１８】一方、ステップＳ１において、ハッシュコマンドでない場合は、ＣＰＵ１は受けた命令が暗号コマンドであるか否かを判断する（Ｓ３）。ここで、暗号コマンドである場合は、暗号認証処理すなわちＥＥＰＲＯＭ４またはＲＡＭ５に格納してあるデータを暗号化して署名データを生成する処理を行う（Ｓ４）。ステップＳ３において、暗号コマンドでない場合、すなわち受けた命令がハッシュコマンドでもなく暗号コマンドでもない場合は不当なコマンドとしてエラーステータスを端末装置側に送る（Ｓ５）。

【００１９】図３は、ＩＣカード１０におけるレコード結合処理及び圧縮処理を示すフローチャートである。以下の処理動作は、主にＣＰＵ１が制御する。まず、作業バッファの先頭アドレスを所定のレジスタにセットする（Ｓ１１）。ここで、作業バッファとは、当該レコード結合処理または圧縮処理を実行する際に用いるメモリ領域であり、本実施形態ではＲＡＭ５またはＥＥＰＲＯＭ４内の一部の領域である。

【００２０】そして、ハッシュ処理すなわち圧縮処理で用いる初期データをWORK1にセットする（Ｓ１２）。ここで、WORK1は、作業バッファの一部であり、前記先頭アドレスに基づいて所定領域が確保されている。次に、レジスタｒ０に、本レコード結合処理の対象となるレコードの数をセットする（Ｓ１３）。本レコード結合処理の対象となるレコードは、例えば、金融取引等における金額、取引時間、取引場所などの履歴を示すデータからなる。そして、各レコードは、図６に示すようなファイル構造で、ＲＡＭ５またはＥＥＰＲＯＭ４に格納されている。

【００２１】次に、レジスタｒ１に、結合処理の対象となるレコードのレコード番号の初期値として「１」をセットする（Ｓ１４）。そして、レジスタｒ２に、作業バッファ内データ長の初期値として、「０」をセットする（Ｓ１５）。ここで、作業バッファ内データ長とは、作業バッファ内に格納した、すなわち作業バッファ内で既に結合したデータの長さをいう。

【００２２】その後、レジスタｒ３に、レジスタｒ１で

指定しているレコードのレコード長（バイト数）をセットする（Ｓ１６）。次に、作業バッファがＥＮＤの位置にきているか、すなわち、作業バッファが結合したデータでいっぱいか否かを判断する（Ｓ１７）。ここで、作業バッファがエンドの位置にきていない場合は、レジスタｒ１が保持している番号のレコードのデータを読み出し、これを作業バッファに格納する（Ｓ１８）。ここで、その読み出し対象のレコードのデータ長（バイト数）はレジスタｒ３に保持されているので、そのレジスタｒ３の値だけ当該レコードのデータを作業バッファに格納すれば、１レコード分のデータが作業バッファに格納される。すなわち、１つのレコードに含まれるデータ全部を読み出して、作業バッファに格納することで、その作業バッファにおいて各レコードを結合している。

【００２３】その後、レジスタｒ２に、レジスタｒ２の内容とレジスタｒ３の内容とを加算した値をセットする。これにより、バッファ内のデータ長がレジスタｒ２にセットされる。さらに、レジスタｒ１の内容を「１」増加させて、新たなレジスタｒ１の内容とする（Ｓ１９）。これにより、作業対象となるレコード番号が「１」増加する。次に、レジスタｒ０の内容を「１」減少させて、新たなレジスタｒ０の内容とする（Ｓ２０）。そして、レジスタｒ０の内容が「０」になったか否かを判断し、「０」でない場合は、まだ結合及び圧縮処理をしていないレコードがあるので、ステップＳ１６に戻る（Ｓ２１）。

【００２４】これらのステップＳ１３、２０、２１の処理によって、本レコード結合処理の対象となるレコードの数だけ、ステップＳ１６から２１の処理が繰り返され、各レコードを結合していく。

【００２５】ステップＳ２１において、レジスタｒ０の内容が「０」になった場合は、図４に示すハッシュ処理をする（２２）。すなわち、レジスタｒ０の内容が「０」になったことで、結合処理の対象となるレコードの全てが作業バッファにおいて結合されたことがわかる。そして、その結合したレコードの全体について、ハッシュ処理を１回行いデータ圧縮する。

【００２６】一方、ステップＳ１７において、作業バッファがいっぱいになった場合は、レジスタｒ３の内容から読み出しバイト数を減算して、新しいレジスタｒ３の内容とする（Ｓ２４）。これは、読み出し作業バッファにｒ３バイトずつ格納していくと、作業バッファの大きさに満たないデータが出てくるため、その満たない大きさのデータが（ｒ３－読み出しバイト数）になり、これが次のデータを入れるバッファｒ３の大きさを示すものである。

【００２７】その後、図４に示すハッシュ処理、すなわち圧縮処理をする（Ｓ２５）。これらは、作業バッファがいっぱいになっり、所定量のデータがその作業バッファにおいて結合されたので、その結合されたデータを圧縮処理するものである。その後、まだハッシュ処理されずに残っているレコードを結合して、圧縮処理をすべくステップＳ１７の処理に移る。

【００２８】図４は、ＩＣカード１０における圧縮処理を示すフローチャートである。すなわち、本フローチャートは、図３におけるステップＳ２２およびステップＳ２４の各圧縮処理を具体的に示したものである。ここで、バッファWORK1,WORK2は、それぞれ圧縮処理についての作業用のバッファであり、それぞれＲＡＭ５内にセットするものとする。なお、その作業用のメモリ領域は、ＥＥＰＲＯＭ４内において確保してもよい。

【００２９】まず、初期値をバッファWORK1にセットする（Ｓ３０）。次に、バッファWORK２に圧縮対象となるデータをセットする（Ｓ３１）。そして、バッファWORK2にあるデータとバッファWORK1にあるデータとを加え合わせ、その加え合わせたデータをハッシュ関数を用いて圧縮する（Ｓ３２）。その圧縮したデータは、バッファWORK1にセットする（Ｓ３３）。また、その圧縮したデータであってバッファWORK1にセットしたデータは、バッファWORK2にあるデータに加え合わせられて、ステップＳ３２における圧縮対象となるデータとなる。その後、作業バッファの先頭アドレスを所定のレジスタにセットし（Ｓ３４）、ハッシュ処理を終了する。

【００３０】なお、圧縮アルゴリズムとしては、一般的に知られている一方向性ハッシュ関数を使用する。その一方向性ハッシュ関数としては、例えば、ＭＤ４、ＭＤ５、ＳＨＡ、ＲＩＰＥ－ＭＤ等を用いることができる。なお、ＭＤ４、ＭＤ５における”ＭＤ”は、「Messeage Digest」の略である。また、”ＳＨＡ”は、「SeureHash Algorithm」の略である。

【００３１】図５は、ＩＣカード１０における暗号化処理を示すフローチャートである。まず、キーデータを読み出す（Ｓ４１）。ここで、キーデータはＥＥＰＲＯＭ４またはＲＯＭ３に記憶されている。次に、指定レコードのアドレスを計算する（Ｓ４２）。次に、ＥＥＰＲＯＭ４またはＲＯＭ３に記憶してある暗号用データのアドレスをセットする（Ｓ４３）。次に、キーデータアドレスをセットする（Ｓ４４）。

【００３２】次に、コマンドで指定されたＮおよびｅ（ｄ）でＲＳＡ（Rivest Shamir Adleman）方式の暗号処理を行う（Ｓ４５）。ＲＳＡ方式とは、公開鍵暗号方式の一つであり、素因数分解が実際上困難である（例えば５１２ビット以上の）大きな整数を法とする剰余べキ乗を用いる方式である。なお、本発明における暗号処理は、ＲＳＡ方式に限定されるものではなく、例えば、ＤＥＳ方式、またはＦＥＡＬ方式などの他の方式を用いることもできる。

【００３３】ここで、暗号化におけるべき乗剰余演算は、コプロセッサ２が行う。これにより、暗号処理が高速に実行される。そして、暗号化処理が完了した後、そ

の処理結果を署名データとしてＥＥＰＲＯＭ４に格納する。これとともに、出力バッファに署名データをセットして端末装置側に送出する（Ｓ４６）。

【００３４】図６は、ＩＣカード１０内における認証用データについてのファイル構造を示す説明図である。まず、ディレクトリ構造としては、データファイルの先頭アドレスと、ＲＡＭ５またはＥＥＰＲＯＭ４に格納されているレコードの数とが格納されている。そして、認証用のデータもディレクトリ構造の一つとして格納されている。一方、圧縮処理および暗号化の対象となる取引データなどは、レコード単位でＲＡＭ５またはＥＥＰＲＯＭ４に格納されている。そして、各レコードには、自身のレコード長がデータとして付加されている。

【００３５】図７は、ＩＣカード１０に対して端末装置側が送出するコマンドフォーマットを示す説明図である。本図では、ハッシュ処理を指令するハッシュコマンドと、暗号化処理を指令する暗号コマンドが示してある。また、各処理が正常に実行されたことを示すレスポンス信号のフォーマットも示してある。

【００３６】これらにより、本ＩＣカードは、複数のレコードからなる取引データをＣＰＵ１が本ＩＣカードの内部で結合し、この結合したデータのかたまり、すなわちブロックについてＩＣカード１０の内部で圧縮処理および暗号化処理をして署名データを生成するので、取引データを外部に出力せずに署名データを生成することができる。したがって、本ＩＣカードによれば、金融取引システム等で高いセキュリティ性が求められる署名データの偽造および改ざんをほとんど不可能にすることができる。

【００３７】また、本ＩＣカードでは、複数のレコードを結合したデータのかたまりについて圧縮処理および暗号化処理をするので、各レコード毎に圧縮処理および暗号化処理をした場合よりもＭＡＣデータ等を少なくすることができる。これは、以下の理由による。各レコード毎に暗号化処理をすると各レコード毎にＭＡＣデータ等が付与される。一方、データのかたまりについて暗号化処理をするとそのかたまりに１個のＭＡＣデータが１つ付与されるだけであるからである。したがって、本ＩＣカードは、そのＩＣカードに内蔵するメモリを効率的に使用することができる。

【００３８】また、本ＩＣカードでは、複数のレコードを結合したデータのかたまりについて圧縮処理および暗号化処理をして署名データを生成するので、各レコード毎に圧縮処理および暗号化処理をして署名データを生成した場合よりも圧縮処理および暗号化処理の回数を大幅に低減することができる。したがって、本ＩＣカードは、署名データの生成を高速に実行することができる。

【００３９】Ｃ：変形例

上述の実施形態においては、複数のレコードを１個のブロックに結合し、そのブロック毎に圧縮処理および暗号化処理をしている。しかし、本発明は上述の実施形態に限定されるものではない。例えば、複数のレコードを１個のブロックに結合し、そのブロックを複数生成し、複数のブロックを１個のブロック集合に結合して、そのブロック集合毎に圧縮処理および暗号化処理を行ってもよい。

【００４０】

【発明の効果】以上説明したように、本発明によれば、複数のレコードを１個のブロックに結合し、さらにそのブロック毎に連続して圧縮してから暗号化して署名データを生成する手段をＩＣカード自身がもつので、取引データを前記レコードに含ませた場合でも、その取引データを外部に出力せずに、ＩＣカードの内部で署名データを生成することができ、セキュリティ性の高いＩＣカードを提供することができる。すなわち、本発明によれば、署名データの偽造および改ざんがほとんど不可能となるＩＣカードを提供することができる。

【００４１】また、本発明によれば、複数のレコードを結合したブロック毎に、圧縮処理および暗号化処理をするので、各レコード毎に圧縮処理および暗号化処理をした場合よりもＭＡＣデータ等を少なくすることができ、ＩＣカードに内蔵するメモリを効率的に使用することができる。

【図面の簡単な説明】

【図１】　本発明の実施形態によるＩＣカードの構成を示すブロック図である。

【図２】　同実施形態におけるコマンド処理を示すフローチャートである。

【図３】　同実施形態におけるレコード結合処理を示すフローチャートである。

【図４】　同実施形態における圧縮処理を示すフローチャートである。

【図５】　同実施形態における暗号化処理を示すフローチャートである。

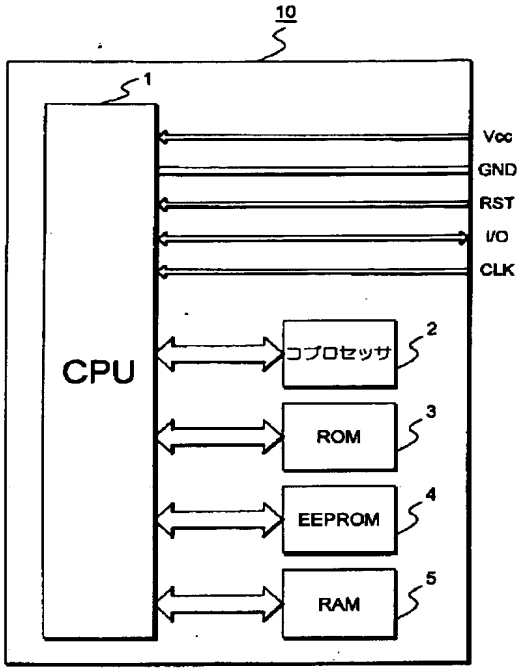
【図６】　同実施形態におけるファイル構造を示す説明図である。

【図７】　同実施形態におけるコマンドフォーマットを示す説明図である。

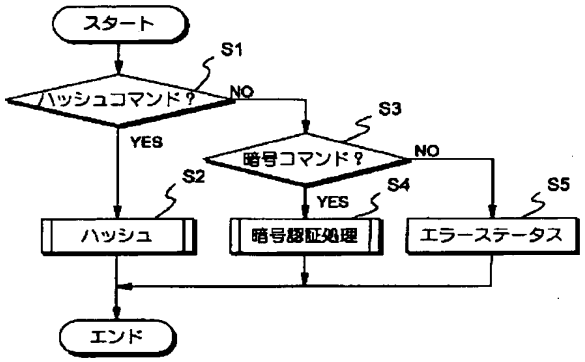
【符号の説明】

１　ＣＰＵ
２　コプロセッサ
３　ＲＯＭ
４　ＥＥＰＲＯＭ
５　ＲＡＭ

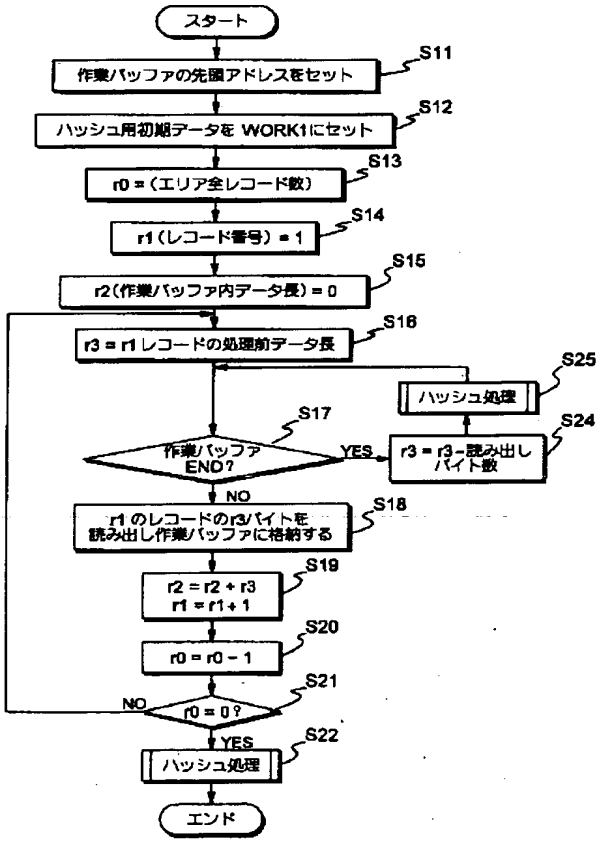
【図１】


10
1
CPU
Vcc
GND
RST
I/O
CLK
コプロセッサ
2
ROM
3
EEPROM
4
RAM
5

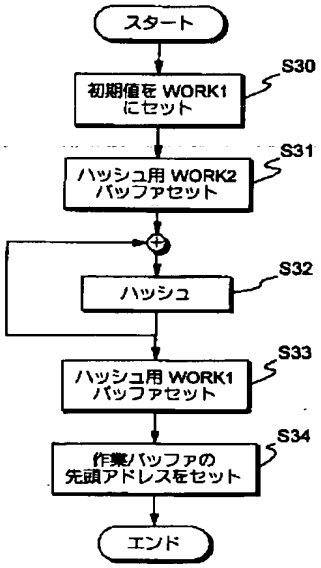
【図２】


スタート
S1
ハッシュコマンド？
NO
YES
S3
暗号コマンド？
NO
YES
S2
S4
S5
ハッシュ
暗号認証処理
エラーステータス
エンド

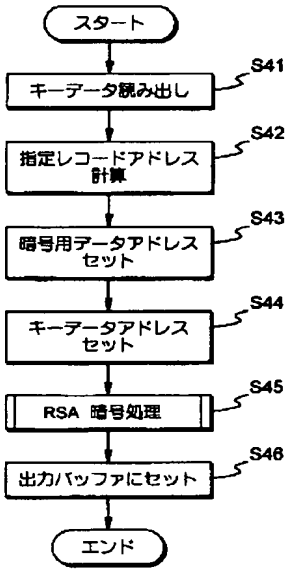
【図３】


スタート
作業バッファの先頭アドレスをセット
S11
ハッシュ用初期データを WORK1にセット
S12
r0 = (エリア全レコード数)
S13
r1 (レコード番号) = 1
S14
r2(作業バッファ内データ長) = 0
S15
r3 = r1 レコードの処理前データ長
S16
S25
ハッシュ処理
S17
作業バッファ END?
YES
r3 = r3 - 読み出しバイト数
S24
NO
r1 のレコードの r3 バイトを読み出し作業バッファに格納する
S18
r2 = r2 + r3
r1 = r1 + 1
S19
r0 = r0 - 1
S20
NO
r0 = 0 ?
S21
YES
ハッシュ処理
S22
エンド

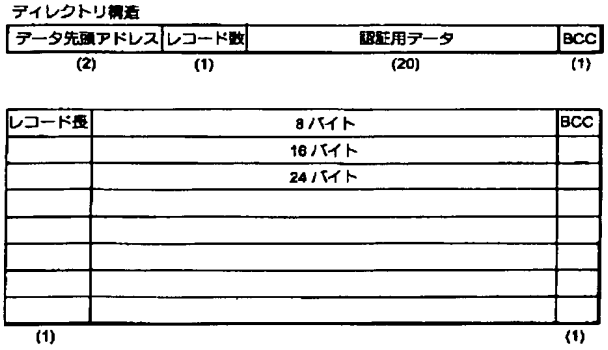
【図４】


スタート
初期値を WORK1 にセット
S30
ハッシュ用 WORK2 バッファセット
S31
ハッシュ
S32
ハッシュ用 WORK1 バッファセット
S33
作業バッファの先頭アドレスをセット
S34
エンド

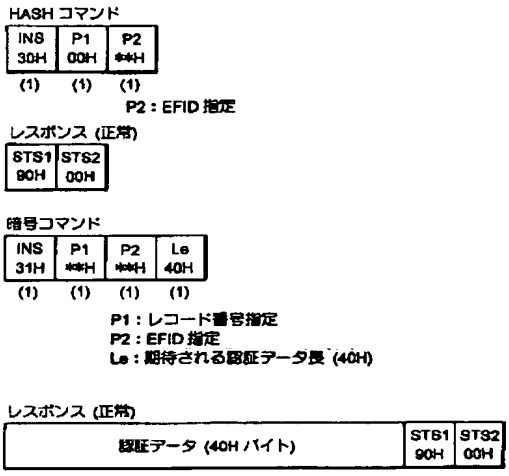
【図５】



【図６】



【図７】



フロントページの続き

(51) Int. Cl.⁶	識別記号	F I	
G09C 1/00	640	G09C 1/00	640B

(72)発明者　高山　文博
東京都台東区台東１丁目５番１号　凸版印刷株式会社内

(72)発明者　大間　康之
東京都新宿区西新宿３丁目19番２号　日本電信電話株式会社内

(72)発明者　八田　義洋
東京都新宿区西新宿３丁目19番２号　日本
電信電話株式会社内

(72)発明者　平田　真一
東京都新宿区西新宿３丁目19番２号　日本
電信電話株式会社内